Supplementary Materials of Decomposed Distribution Matching in Dataset Condensation

Sahar Rahimi Malakshan, Mohammad Saeed Ebrahimi Saadabadi,
Ali Dabouei, and Nasser M. Nasrabadi
sr00033, me00018, Ad0046@mix.wvu.edu, nasser.nasrabadi@mail.wvu.edu

A Impact of Style Discrepancy on DC

To illustrate the effect of style discrepancy between the condensed and original datasets, we conduct experiments in which we drift the style of samples from Herding [45] coreset selection $(\boldsymbol{\mu}^l, \boldsymbol{\sigma}^l)$ toward that of DM $(\widehat{\boldsymbol{\mu}}^l, \widehat{\boldsymbol{\sigma}}^l)$, as shown in Figure 1.c, and d of the manuscript. Specifically, during the training of a CNN, the drifted style information is computed by a convex combination of $(\boldsymbol{\mu}^l, \boldsymbol{\sigma}^l)$ and $(\widehat{\boldsymbol{\mu}}^l, \widehat{\boldsymbol{\sigma}}^l)$:

$$\boldsymbol{\sigma}^l_{drifted} = (1-\gamma)\boldsymbol{\sigma}^l + \gamma\widehat{\boldsymbol{\sigma}}^l, \tag{12}$$

$$\boldsymbol{\mu}^l{}_{drifted} = (1-\gamma)\boldsymbol{\mu}^l + \gamma\widehat{\boldsymbol{\mu}}^l, \tag{13}$$

where γ denotes the drift ratio, *i.e.*, the extent to which the style information shifts from the original towards the target style. Then, we compute the feature maps with the drifted style information, following the approach of the pioneering work [56]:

$$\boldsymbol{\Phi}^l_{drifted} = \sqrt{\boldsymbol{\sigma}^l{}_{drifted}}\frac{\boldsymbol{\Phi}^l - \boldsymbol{\mu}^l}{\sqrt{\boldsymbol{\sigma}^l}} + \boldsymbol{\mu}^l{}_{drifted}. \tag{14}$$

Subsequently, $\boldsymbol{\Phi}^l_{drifted}$ passes through the remaining layers of $\boldsymbol{\Phi}$, as shown in Figure 5a.

Figures 1.c, and d of the manuscript show the effect of style discrepancy. As the style diverges from that of the original samples, *i.e.*, increasing the gap between the training and testing data styles, the model performance decreases. This outcome is consistent with the well-established style bias in DNNs [19, 2, 72, 65, 65].

B Ablation on Hyperparamers

B.1 α in Equation 7

The overall style matching objective is defined as $L_S = \alpha L_{MM} + L_{CM}$, where α is a weighting factor balancing the moments matching, L_{MM}, and correlation matching, L_{CM}, losses. Here, we perform ablation on the α, shown in Figure 5b, and c. Results show that employing both L_{MM} and L_{CM} with equal weight, *i.e.*, $\alpha = 1$, yields the best performance, highlighting the complementary roles of these two losses. Specifically, L_{MM} captures style information represented by the mean and variance of feature maps, while L_{CM} captures style information through the correlation among feature maps.

	k										
$IPC\times$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
IPC=10	48.95	49.15	49.90	49.83	49.42	48.81	48.14	47.85	47.54	46.65	45.20
IPC=50	63.00	63.56	63.96	63.68	63.45	63.14	62.5	61.9	61.2	61.15	58.5

Table 6. Ablation study on the hyperparameter k for L_{ICD} in Equation 9 for IPC=10 and 50 on CIFAR10 dataset, showing the testing accuracy (%) of the condensed dataset on CIFAR10.

B.2 k in Equation 9

Figures 4a, and b of the manuscript show that condensed samples learned by DM [67] tend to form dense clusters, indicating the need for a criterion to encourage diversity. In L_{ICD}, k specifies the number of nearest intra-class samples in the embedding space. We designed the loss to repel each condensed sample from its k closest intra-class neighbors, thereby enhancing intra-class diversity. We conducted experiments to determine the optimal k for different IPCs. A smaller k focuses on diversifying a localized neighborhood of samples, while a larger k degrades results by encouraging broader dispersion. Large k values can overly disperse synthetic samples, compromising class consistency and authenticity. Our experiments revealed that setting k to $0.2 \times$ IPC yields optimal results for both IPC=10 and IPC=50.

B.3 β in Equation 10 and λ in Equation 11

Figure 6 illustrates the impact of β and λ on our method's performance, corresponding to L_{ICD} and L_S in Equations 10 and 11, respectively. Optimal results for both loss components are achieved at $\beta = 10$ and $\lambda = 5 \times 10^3$. The magnitudes of L_{ICD} and L_S are significantly lower compared to L_{MMD}, necessitating the adjustment of hyperparameters to higher values for balance. Results in Figure 6

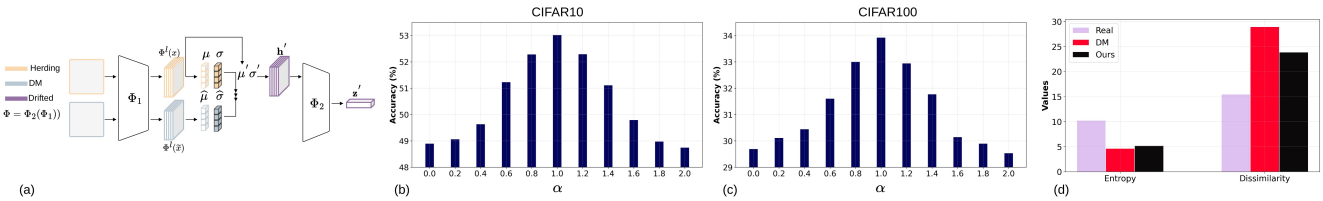

Figure 5. a) Details of the experiment in Figure 1c, and d of the manuscript. (b, c) Ablation on α in Equation 7 for IPC=10 on both CIFAR10 and CIFAR100 datasets. d) Average dissimilarity and entropy texture features based on GLCM method [75] across real and condensed set with IPC=10 for one category in CIFAR10 datasets. The texture features of the condensed set learned by our method more closely resemble those of real images, compared to the DM method.

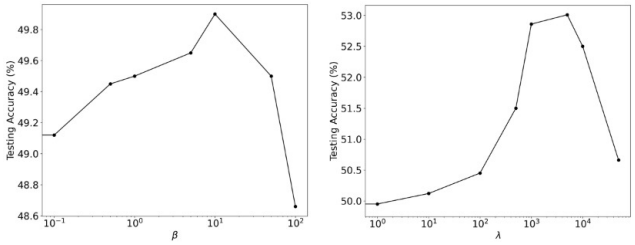

Figure 6. Ablation on β and λ in Equations 10 and 11 of the manuscript, respectively, for IPC=10 on CIFAR10 dataset.

demonstrate that integrating style information and promoting intra-class diversity consistently enhances performance, up to a threshold of 5×10^3 and 10, respectively. Beyond this point, performance starts to decline, attributed to an overemphasis on style matching at the expense of the discriminative features highlighted by L_{MMD}. Moreover, it is vital to balance intra-class diversity enhancement to prevent class overlap or confusion. Therefore, exceeding the optimal thresholds for the style-matching and intra-class diversity coefficients results in a decline in model performance.

C Visualization

Figures 7 and 8 display the resulting condensed sets for CIFAR100 and TinyImageNet, learned by DM and our method, alongside the real images. The improvement in visual quality and diversity with our method is attributed to the SM module and ICD component, detailed in Sections 3.3 and 3.4 of the manuscript, which effectively reduce the style gap between original and condensed sets and enhance intra-class diversity among condensed samples, respectively.

D Style

D.1 Style Gap Analysis

As discussed in the Introduction, our comparison of style indicators between CIFAR10's real and condensed datasets (Figure 1.a) reveals a significant style gap. To evaluate our method's effectiveness in mitigating this gap, we repeated the experiment with our approach, as shown in Figure 9. The results demonstrate that our method successfully narrows the style discrepancy using the SM module.

D.2 Texture Analysis

Conventionally, style can be characterized by the textural attributes of an image, which include roughness, smoothness, and color diversity in the image [16, 18]. Texture analysis in the field of image processing is a crucial component and can be broadly categorized into four main approaches: statistical, geometric, model-based, and signal processing techniques [74, 75]. Among these, the Gray-Level Co-occurrence Matrix (GLCM), introduced by Haralick *et al.* [75], is a prominent statistical method for texture analysis. GLCM is foundational for texture analysis, emphasizing the spatial distribution and relation of pixels to describe an image's surface characteristics effectively [75, 77].

Utilizing the GLCM method, we employ two texture features including dissimilarity and entropy to analyze the textural statistics of images, which are computed as [76, 77]:

$$\text{Dissimilarity} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p(i,j) \cdot |i-j|, \tag{15}$$

$$\text{Entropy} = -\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p(i,j) \cdot \ln(p(i,j)), \tag{16}$$

where n denotes the grayscale level, and $p(i,j)$ is the normalized grayscale value at positions i and j within the kernel, summing to 1. We employ different kernels (3×3 and 5×5, a region or a set of neighbors around a central pixel) and report the average of them in the results. Dissimilarity evaluates the variation in intensity among adjacent pixel pairs, offering insights into texture contrast and complexity [76]. Entropy, measures the randomness in intensity distribution, thereby reflecting the unpredictability and diversity of textural patterns [76].

As illustrated in Figure 5d, there is a significant gap in both texture features between real images and those learned

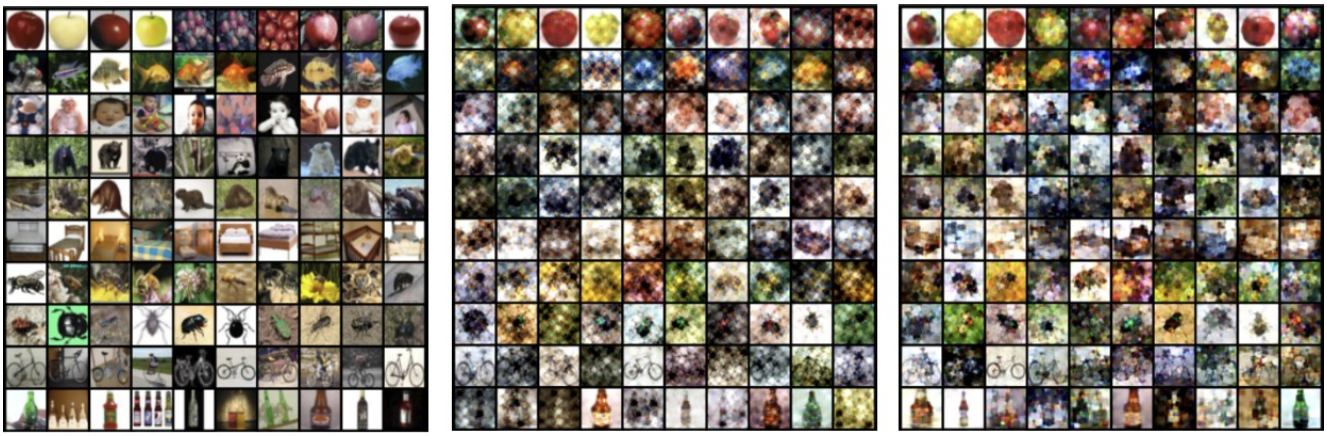

Figure 7. Visualizations of (a) real and (b) condensed images learned by DM and (c) our method for CIFAR100 with IPC=10. Both methods are initialized from real samples. Our method exhibits improved visual quality and diversity compared to DM.

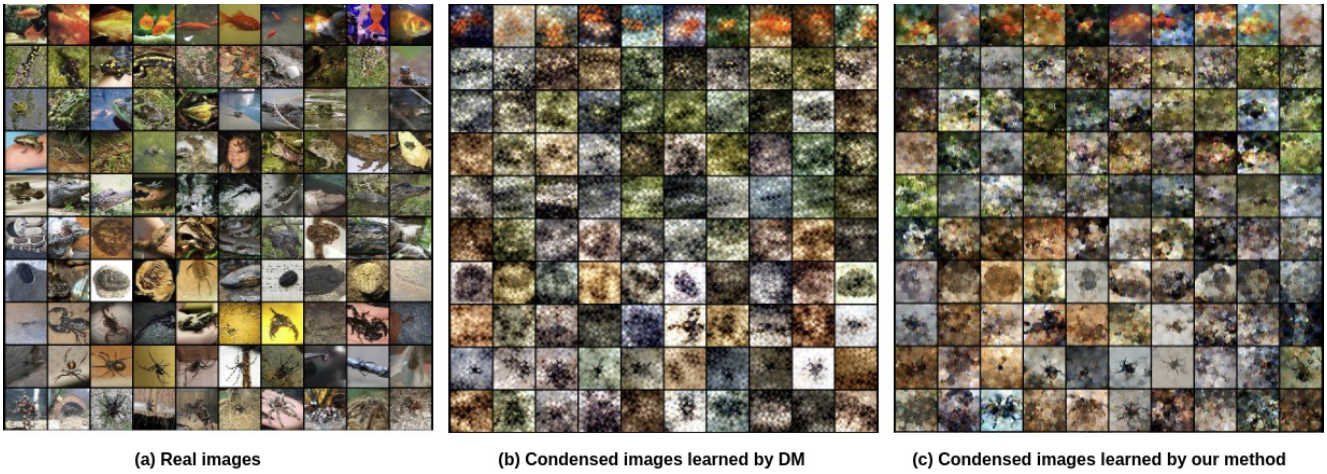

Figure 8. Visualizations of (a) real and (b) condensed images learned by DM and (c) our method for TinyImageNet with IPC=10. Both methods are initialized from real samples. Our method exhibits improved visual quality and diversity compared to DM.

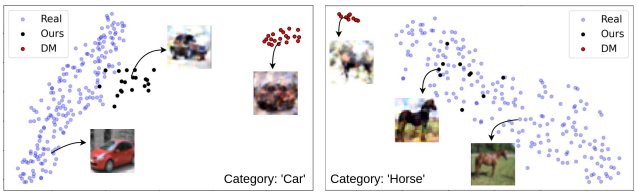

Figure 9. 2D t-SNE visualization of style statistics computed from the first layer's feature map of ConvNet, for real CIFAR10 images, and condensed set learned by DM and our method for two categories, demonstrating the effectiveness of our approach in reducing the style gap.

by the DM. The usage of the style matching module introduced by our method brings the texture features in the condensed set closer to real data compared to the baseline of DM [67], as shown in Figure 5d. Specifically, our method achieves dissimilarity and entropy features that are 5% and 0.56% closer to real features compared to DM, respectively, indicating improvements in texture matching between original and learned condensed sets in our method.

E Style Matching in Multiple Layers

To evaluate the impact of the SM module across different blocks, we applied it to each block of the ConvNet architecture, which consists of three convolutional blocks. Our results, presented in Table 8, indicate that applying this module individually after each block improves performance. These consistent enhancements across different blocks highlight the presence of beneficial style knowledge for DC at various depths within the DNN. Ultimately, ap-

plying this module across all three blocks yields the best results, as demonstrated in Table 8, underscoring the existence of distinct style information throughout the layers of the DNN.

F Application: Neural Architecture Search

Neural Architecture Search (NAS) aims to identify the best DNN architecture candidates. NAS has become an important use case for dataset condensation (DC) since a condensed dataset can be used as a proxy for the original data to efficiently search for optimal architectures. Here, we compare the performance of the proposed method with three baselines: DM, DSA, and Random Selection. Following [68], we explore the application of our method in NAS on the CIFAR-10 dataset, using a search space of 720 ConvNets by varying hyperparameters. Please refer to [68] for full experimental details. We trained architectures on both the original and condensed datasets for 200 epochs. Table 7 presents: 1) accuracy on the test data, 2) Spearman's rank correlation coefficient between the testing accuracy of the top models selected using condensed datasets and the whole training data, 3) training time required for training 720 architectures, and 4) memory footprint of the datasets. The proposed method achieves the highest accuracy among its competitors, coming within one percent of the accuracy obtained by training on the full CIFAR-10 dataset. Moreover, the training time is significantly reduced from 8604.3 minutes to 142.6 minutes. Additionally, our method enhances the Spearman's rank correlation coefficient for DM, indicating that a reliable ranking of architectures is obtained using the proposed method.

	Random	DSA	DM	Ours	Whole Dataset
Accuracy	84.0	82.6	82.8	**84.2**	**85.9**
Correlation	-0.04	0.68	0.76	**0.80**	1.0
Time cost (min)	**142.6**	**142.6**	**142.6**	**142.6**	3580.2
Storage (imgs)	**500**	**500**	**500**	**500**	50000

Table 7. Neural architecture search experiments on CIFAR-10 dataset for the search space of 720 ConvNets.

Block 1	Block 2	Block 3	Accuracy
-	-	-	48.9 ± 0.6
✓	-	-	50.93 ± 0.66
-	✓	-	51.60 ± 0.57
-	-	✓	51.91 ± 0.56
✓	✓	✓	52.29 ± 0.42

Table 8. Ablation on SM module Across ConvNet convolutional blocks for CIFAR10 dataset with IPC=10.

References

[74] CH Chen and David Stork. Handbook of pattern recognition & computer vision. *International Journal of Neural Systems*, 5(3):257, 1994. 2

[75] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973. 2

[76] Geun-Ho Kwak and No-Wook Park. Impact of texture information on crop classification with machine learning and uav images. *Applied Sciences*, 9(4):643, 2019. 2

[77] Zoltan Szantoi, Francisco Escobedo, Amr Abd-Elrahman, Scot Smith, and Leonard Pearlstine. Analyzing fine-scale wetland composition using high resolution imagery and texture features. *International Journal of Applied Earth Observation and Geoinformation*, 23:204–212, 2013. 2